AA

(12) UK Patent Application (19) GB (11) 2 307 319 (13) A

(43) Date of A Publication 21.05.1997

(21) Application No 9623913.2

(22) Date of Filing 18.11.1996

(30) Priority Data
(31) 95041877 (32) 17.11.1995 (33) KR

(71) Applicant(s)
Hyundai Electronics Industries Co., Ltd.

(Incorporated in the Republic of Korea)

San 136-1,Ami-ri, Bubal-eup, Ichon-shi, Kyoungkido 467-860, Republic of Korea

(72) Inventor(s)
Han Heung Kim

(74) Agent and/or Address for Service
John Orchard & Co
Staple Inn Buildings North, High Holborn, LONDON, WC1V 7PZ, United Kingdom

(51) INT CL[6]
G06F 12/08 12/10

(52) UK CL (Edition O)
G4A AMC

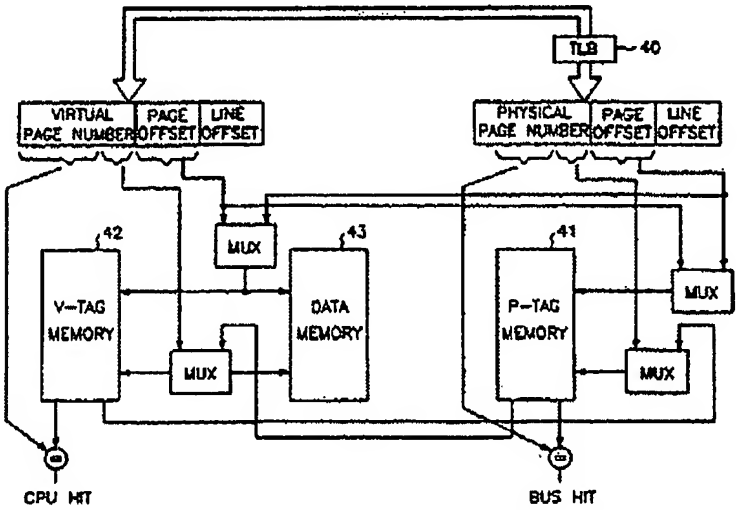
(56) Documents Cited
GB 2280429 A EP 0651332 A1 EP 0369117 A2

(58) Field of Search
UK CL (Edition O) G4A AMC
INT CL[6] G06F 12/08 12/10
Online: WPI

(54) **Dual-directory virtual cache**

(57) This dual-directory virtual cache can be accessed directly using a virtual address without performing an address conversion operation. A virtual tag memory 42 is accessed by a request from a microprocessor in parallel with the data memory 43. A physical tag memory 41 is coupled to a translation look-aside buffer 40, which translates the virtual address into a physical address corresponding to the physical address tag, and a bus monitoring logic circuit. If an "invalid miss" as opposed to a "no-match miss" occurs in the virtual tag memory then the physical tag memory is examined and state information can be copied to the virtual tag memory.

FIG. 4



GB 2 307 319 A

FIG. 1

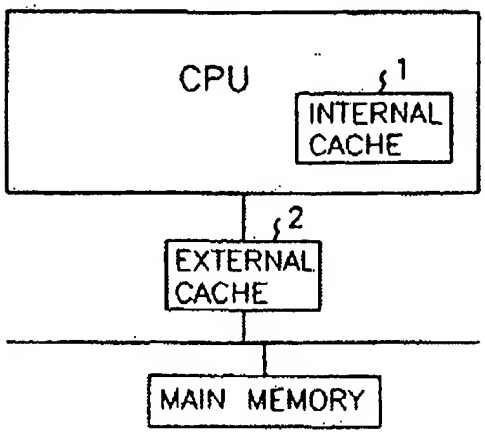

CPU
1
INTERNAL CACHE
2
EXTERNAL CACHE
MAIN MEMORY

FIG. 2

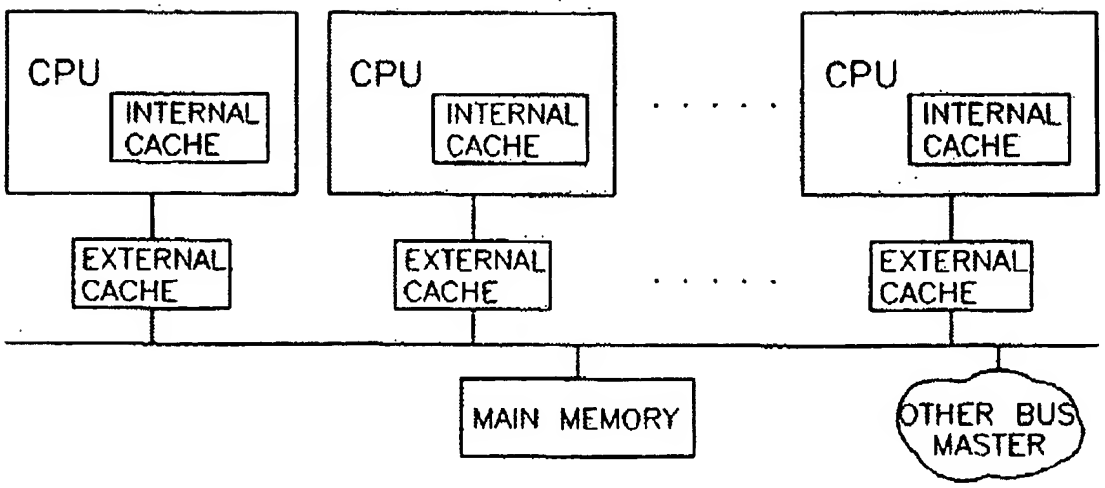

CPU
INTERNAL CACHE
CPU
INTERNAL CACHE
CPU
INTERNAL CACHE
EXTERNAL CACHE
EXTERNAL CACHE
EXTERNAL CACHE
MAIN MEMORY
OTHER BUS MASTER

FIG. 3

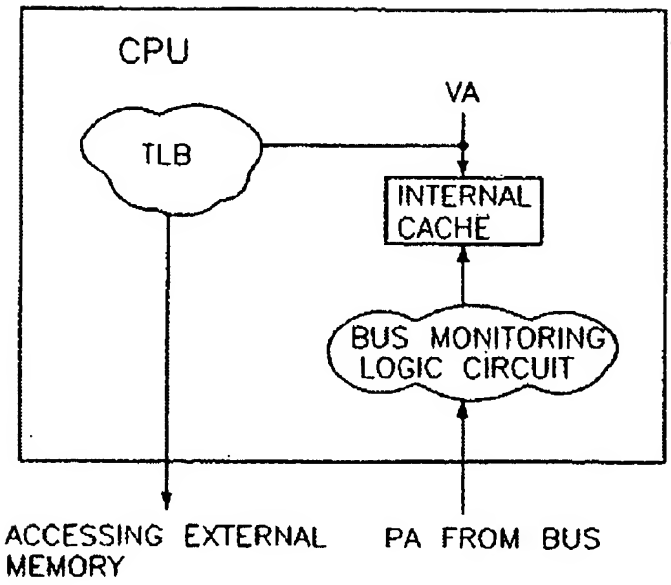

CPU
VA
TLB
INTERNAL CACHE
BUS MONITORING LOGIC CIRCUIT
ACCESSING EXTERNAL MEMORY
PA FROM BUS

FIG. 4

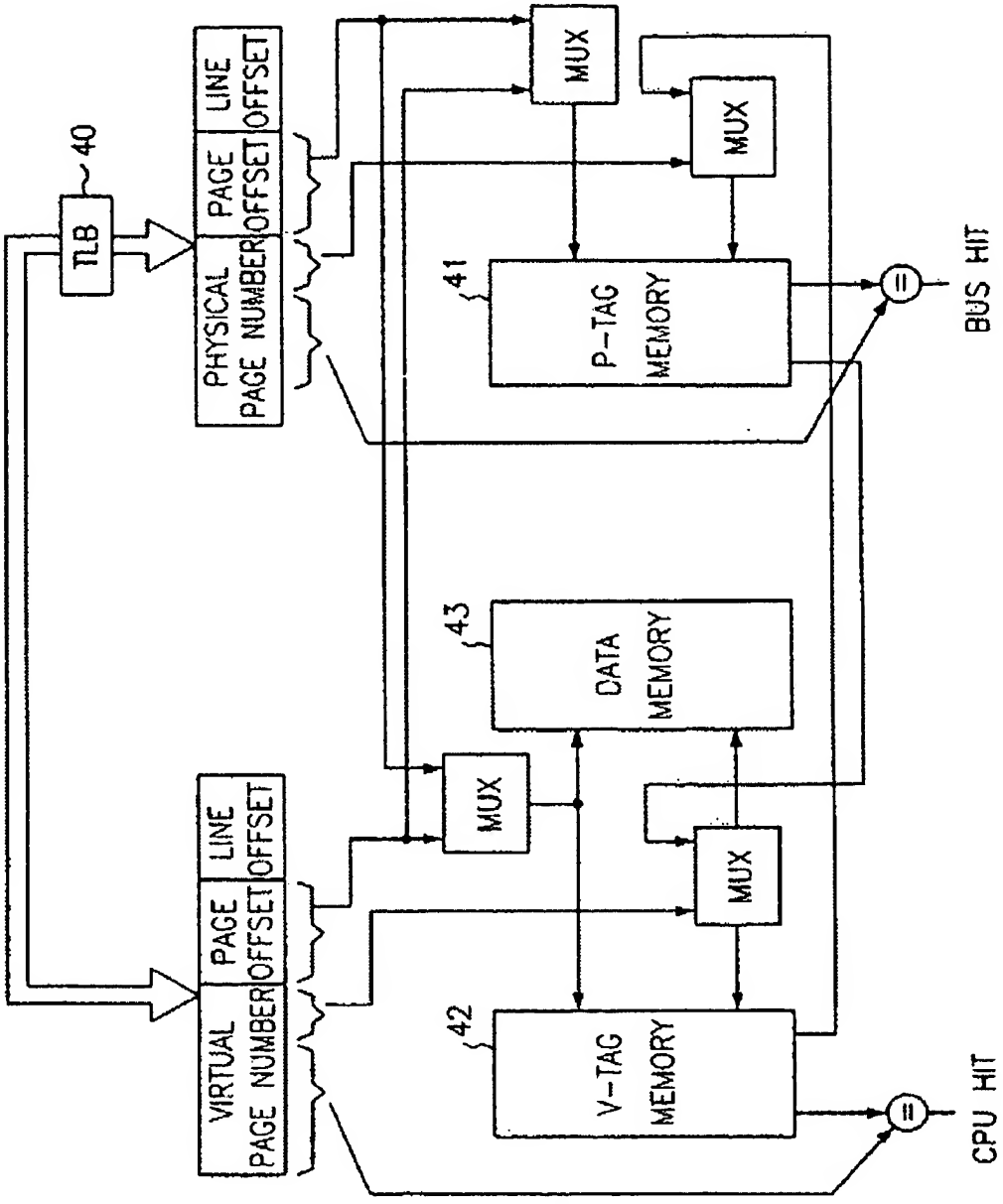

TLB
40
PHYSICAL PAGE NUMBER
PAGE OFFSET
LINE OFFSET
VIRTUAL PAGE NUMBER
PAGE OFFSET
LINE OFFSET
MUX
MUX
MUX
MUX
41
P-TAG MEMORY
42
V-TAG MEMORY
43
DATA MEMORY
BUS HIT
CPU HIT

FIG. 5

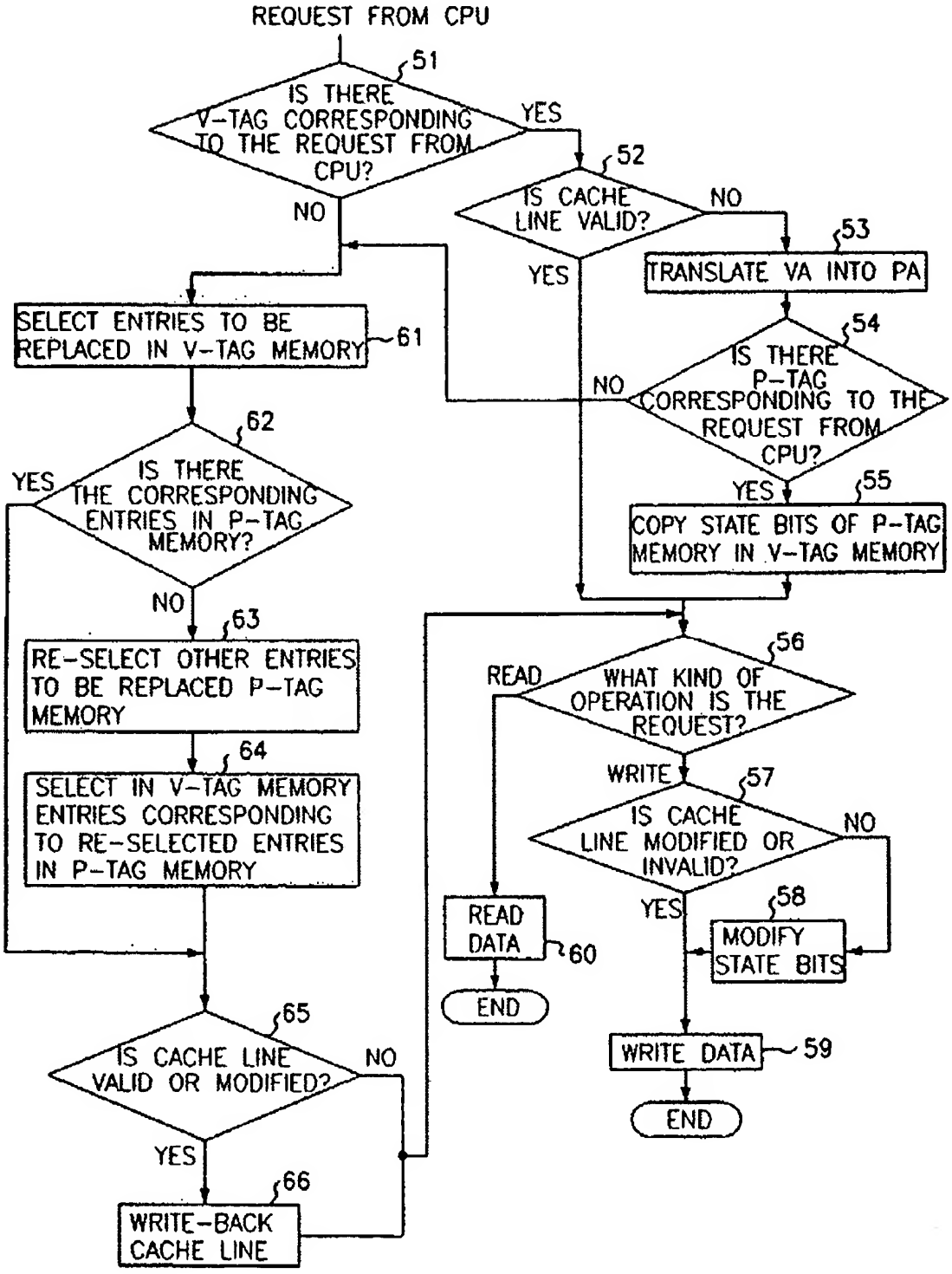
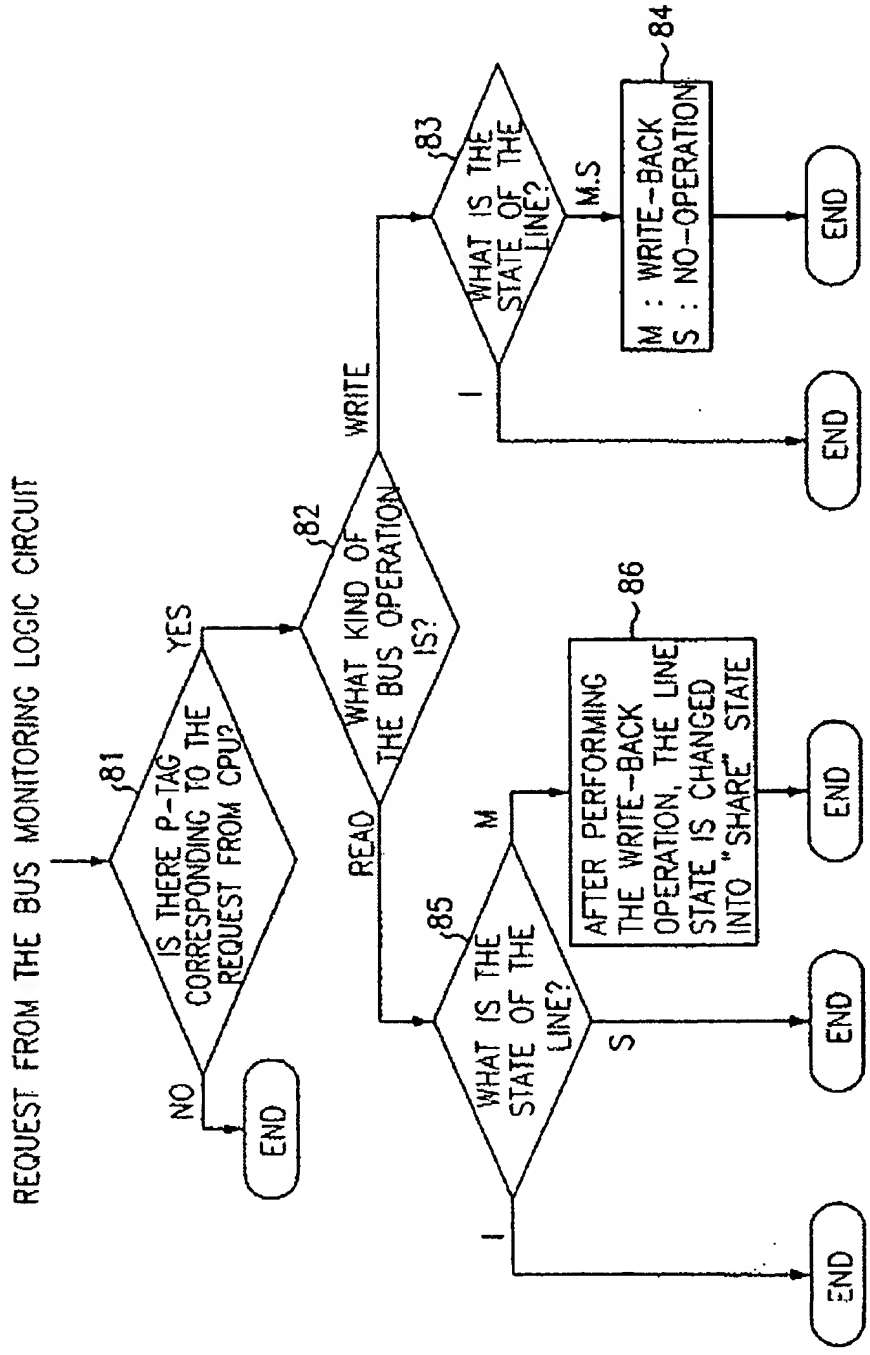

REQUEST FROM CPU
51
IS THERE V-TAG CORRESPONDING TO THE REQUEST FROM CPU?
YES
NO
52
IS CACHE LINE VALID?
NO
YES
53
TRANSLATE VA INTO PA
54
IS THERE P-TAG CORRESPONDING TO THE REQUEST FROM CPU?
NO
YES
55
COPY STATE BITS OF P-TAG MEMORY IN V-TAG MEMORY
SELECT ENTRIES TO BE REPLACED IN V-TAG MEMORY
61
62
IS THERE THE CORRESPONDING ENTRIES IN P-TAG MEMORY?
YES
NO
63
RE-SELECT OTHER ENTRIES TO BE REPLACED P-TAG MEMORY
64
SELECT IN V-TAG MEMORY ENTRIES CORRESPONDING TO RE-SELECTED ENTRIES IN P-TAG MEMORY
65
IS CACHE LINE VALID OR MODIFIED?
NO
YES
66
WRITE-BACK CACHE LINE
56
WHAT KIND OF OPERATION IS THE REQUEST?
READ
WRITE
57
IS CACHE LINE MODIFIED OR INVALID?
NO
YES
58
MODIFY STATE BITS
READ DATA
60
END
WRITE DATA
59
END

FIG. 6


REQUEST FROM THE BUS MONITORING LOGIC CIRCUIT
81
IS THERE P-TAG CORRESPONDING TO THE REQUEST FROM CPU?
YES
NO
END
82
WHAT KIND OF THE BUS OPERATION IS?
WRITE
READ
83
WHAT IS THE STATE OF THE LINE?
M.S
I
84
M : WRITE-BACK
S : NO-OPERATION
END
END
85
WHAT IS THE STATE OF THE LINE?
M
S
I
86
AFTER PERFORMING THE WRITE-BACK OPERATION, THE LINE STATE IS CHANGED INTO "SHARE" STATE
END
END
END

2307319

DUAL-DIRECTORY VIRTUAL CACHE MEMORY AND METHOD FOR CONTROL THEREOF

The present invention relates to a dual-directory virtual cache memory which can be accessed directly using a virtual address without performing an address conversion operation, and to a method for controlling the same.

A cache system has an important effect on the improvement in performance of a microprocessor. In particular, the use of a virtual cache is growing more important day by day to reduce the time required for converting a virtual address into a physical address, and accessing the cache using the converted physical address. Such a virtual cache can be used as an internal storage unit in high-performance systems, more particularly a high-performance microprocessor.

Generally, caches are used to improve the performance of microprocessors. In particular, a virtual cache is used to increase the efficiency of a data storage unit. As a result, a virtual address is used internally, whereas it is converted into a physical address for accessing an external memory.

Also, an internal or an external cache is usually accessed using the converted physical address.

It is natural that the external cache should be accessed using the physical address. However, in the case where the internal cache is to be accessed using the

physical address, an address conversion operation must be performed, resulting in an increase in access time.

In order to solve the above-mentioned problem, the internal cache has been used as a virtual cache which can be accessed directly using a virtual address. In such a virtual cache, internal data consistency must be secured. Also, the internal cache must be externally "snooped", according to a variation in the external cache or memory.

Fig. 1 of the accompanying drawings is a block diagram illustrating the construction of a previously proposed microprocessor system. As shown in this drawing, the previously proposed microprocessor system utilizes a two-level cache system with internal and external caches 1 and 2, for the improvement in the performance thereof. As the external cache 1, a memory is used which is relatively low in cost and large in capacity, but which has a longer access time than the internal cache 1. As the internal cache 1, a memory is used which is high in cost per unit, but which performs a high-speed operation to act up to the high operational speed of a processor (CPU).

Fig. 2 of the accompanying drawings is a block diagram illustrating one example of a multi-processor system using the processor in Fig. 1. As shown in this drawing, the multi-processor system has to consider even the operation of other processors or bus masters. To this end, the multi-processor system requires an external bus to be monitored. Such an operation is typically

called "bus snooping" or simply "snooping". The snooping is an important part of the data consistency in a multiprocessor system.

Fig. 3 of the accompanying drawings is a view illustrating a previously proposed conception of accessing a virtual cache using a virtual address VA and a physical address PA, in a processor employing the virtual cache as an internal cache. As shown in this drawing, a system employing a virtual memory uses the virtual address VA. The access time in the case of accessing the internal cache directly using the virtual address VA is considerably reduced as compared with that in the case of converting the virtual address VA into the physical address PA and accessing the internal cache using the converted physical address PA. However, in the case of accessing the internal cache directly using the virtual address VA, an operation is performed on an external bus according to the physical address PA. As a result, bus snooping becomes an issue.

Features of a dual-directory virtual cache to be described below, by way of example, in illustration of the invention are that it is capable of reducing access time to enhance the performance of a microprocessor, of allowing the associated system to employ a physical cache, and of providing a method for the control thereof.

In a particular embodiment to be described below, by way of example, in illustration of the present invention, there is a cache memory accessed by a virtual address

from a microprocessor, the cache memory including a data memory storing data in cache lines, a virtual tag memory storing a virtual address tag, wherein the virtual tag memory is accessed by a request from a microprocessor in parallel with the data memory, a physical tag memory storing a physical address tag, wherein the physical tag memory is coupled to an address translation means and a bus monitoring logic circuit, and wherein the address translation means translates the virtual address into a physical address corresponding to the physical address tag, and a pointer associating the virtual address tag with the physical address tag.

In another aspect to be described below by way of example, in illustration of the present invention, there is provided a method for accessing a cache memory using a virtual address in response to a request from a microprocessor including a bus monitoring logic circuit, including the steps of determining whether the virtual tag corresponding to the virtual address is stored in a first memory, so that either a hit signal or a miss signal is generated, determining whether a state of a cache line in a first memory storing data is valid in the case where the hit signal is generated, performing a read or a write operation in the case where the state of the cache line is valid, translating the virtual address into a physical address in the case where the state of the cache line is invalid, determining whether a physical tag corresponding to the physical address is stored in a

second memory, copying the state bits of the physical tag into the first memory, if the corresponding physical tag exists in the second memory, and performing a read or a write operation, selecting entries to be replaced in the first memory in the case where there is not a corresponding physical address in the second memory, or the miss signal is generated, determining whether there are entries in the second memory, which correspond to the entries to be replaced in the first memory, and, if there are not, the entries to be replaced in the second memory, re-selecting other entries to be replaced in the second memory and selecting the corresponding entries to be replaced in the first memory according to the re-selected entries, and reading a new cache line from the external memory and performing a read or a write operation.

Arrangements for use in illustrating the invention will now be described, by way of example, with reference to Figs. 4 to 6 of the accompanying drawings, in which:

Fig. 4 is a block diagram illustrating the construction of a dual-directory virtual cache,

Fig. 5 is a flowchart illustrating the operation of the dual-directory virtual cache shown in Fig. 4 which is performed in response to a request from a processor, and

Fig. 6 is a flowchart illustrating the operation of the dual-directory virtual cache shown in Fig. 4 which is performed in response to a request from a bus monitoring logic circuit.

First, Fig. 4 is a schematic diagram showing an

internal cache. As shown in Fig. 4, the internal cache may be classified into three parts. That is, the internal cache memory is a dual-directory virtual cache system including a P-tag memory 41, a V-tag memory 42 and a data memory 43.

A CPU normally accesses the V-tag memory 42 using virtual addresses, and a bus monitoring logic in the CPU accesses the P-tag memory 41 using physical addresses in order to perform a bus snooping operation. The data memory having actual data is accessed by the CPU through the virtual address in parallel with the V-tag memory 42.

The P-tag memory 41, which takes charge of the bus snooping operation, may compare the address in the internal cache with addresses on the bus as quickly as possible, and then access the data. The V-tag memory 42 is accessed by the CPU, together with the data memory 43, so as to respond quickly to its request.

The P-tag memory 41 has a tag for each cache line in the data memory 43 (the cache line is the size of the data which is transmitted between the cache and a main memory at one time, and the tag is data information to identify whether the cache data corresponds to the address, and typically the tag consists of high level addresses). Each tag has unique information which corresponds to the data in the data memory 43. Since P-tag memory 41 includes a table of the pointer for the data memory 43, it has the advantage in that there is no need for it to have association with the V-tag memory 42

and the data memory 43.

The true state of the data for each line of the cache is maintained in the P-tag memory 41. Also, the V-tag memory 42 maintains a little information for each line. However, valid data can be considered as a invalid data in the V-tag memory 42, because the item in the V-tag memory 42 is made invalid in the case where a specific event, such as a task switch, or a change of page table, takes place. This means that the address mapping from the virtual address to the physical address is invalid. After accessing the data, the CPU must perform a verification process.

Therefore, two types of misses may be created in the V-tag memory. One of them is a "no-match miss" as a result of the absence of data. The other is an "invalid miss" necessary to map the virtual address to a new physical address, although the data is valid. Also, each tag memory 41 and 42 has pointer information to connect the virtual address to the physical address.

In the arrangement described, the CPU requests access to the V-tag memory 42 and to the data memory 43. If an invalid miss occurs, an address translation is carried out from a virtual address to a physical address. The translated physical address goes to the P-tag memory 41. If it is proved that there is corresponding data in the data memory 43, the information state in the P-tag memory 41 is copied into the specific location in the V-tag memory 42. By doing so, the "invalid miss" is

processed like a "hit".

When a "no-match miss" occurs, the physical address is obtained through the address translation, and after selecting a replacement target, the corresponding external memory operation takes place.

The snooping (monitoring) logic circuit compares all address on the bus with the P-tag memory 41. At this time, in the case where there is an address corresponding to the P-tag memory 41, different events can occur in the snooping logic circuit as follows:

1) It makes the data invalid and makes the item in the P-tag memory 41 and V-tag memory 42 invalid.

2) In the case where it should drive the data into the exterior, it drives the data into the exterior using the pointer in the P-tag memory 41, and, if necessary, it changes the state of the cache line to be associated with the drive.

3) It reads data on the bus so that it updates the data memory, changing the state of the cache line.

The dual-directory virtual cache will now be illustrated with reference to Figs. 5 and 6.

First, Fig. 5 is a flowchart illustrating the operation of the dual-directory virtual cache in Fig. 4 which is carried out in response to a request from a processor.

First, the CPU determines whether the virtual address, in response to the request, is stored in the V-tag memory. That is, "hit/miss" is inspected using the

virtual address, at step 51. If the virtual address corresponding to the request from the CPU is stored in the V-tag memory, a hit signal is generated so that the state of the corresponding cache line is inspected at step 52. In the case where the state of the cache line is valid, the conventional operation (write or read), according to the request, is performed at steps 56 to 60. In the case where the state of the cache line is invalid, the virtual address is, at step 53, translated into a physical address by a translation look-aside buffer (TLB) so that the translated physical address is inspected in the P-tag memory at step 54.

If a corresponding physical tag exists in the P-tag memory, the state information, which the corresponding P-tag indicates, is copied into the V-tag entry at step 55, and the write or read operation is performed at steps 56, as described above.

In the case where there is not a corresponding physical tag in the P-tag memory at step 54, or the miss signal is generated at step 51, the PU selects, at step 61, entries to be replaced in the V-tag memory, and determines whether there are corresponding (associated by a pointer) entries in the P-tag memory, or not, at step 62. If there are corresponding entries in the P-tag memory, the CPU, at step 65, examines whether the cache line is valid and the cache line has been modified. In the case where the cache line is valid and has been modified, the CPU writes the corresponding cache line in

an external memory at step 66.

At step 62, if there are not corresponding entries in the P-tag memory, the CPU re-selects other entries to be replaced in the P-tag memory at step 63 and, at step 64, selects, in the V-tag memory, the V-tag entries corresponding to the selected P-tag entries, and performs steps 65 and 66 with respect to the selected V-tag entries. By doing so, the CPU updates the V-tag and P-tag memories, and carries out the read or write operation using the external data at steps 56 to 60.

Fig. 6 is a flowchart illustrating the operation of the dual-directory virtual cache shown in Fig. 4 which is performed in response to a request from a bus monitoring logic circuit.

The bus monitoring logic circuit examines whether there is a tag corresponding to the input physical address signals in the P-tag memory at step 81. If there is a corresponding tag in the P-tag memory, the bus monitoring logic circuit examines the bus operation at step 82.

In the case of a read operation, according to the state of the cache line, such as "invalid (I)", "modification (M)" and "share (S)", the data stored in the cache memory may be provided to the external bus and the state of the provided cache line, which is varied as a cache memory read operation, is typically performed by the conventional monitoring logic circuit. That is to say, if the state of the requested cache line is

"modification (M)" the data is written in the external memory and the state of the requested cache line is changed into the "share (S)" state at steps 85 and 86.

In the case of a write operation, the data stored in the cache memory should be made invalid. However, the write-back operation may be performed according to the state of the cache line at step 83. In particular, in the case where the state of the cache line is the "modification" state, the corresponding data should be write back in external memory and then the state is changed into the "invalid" state because the cache line having the "modification" state has the updated information.

Furthermore, when the entries in the P-tag memory are in the "invalid" state, the corresponding entries in the V-tag memory must also be brought into the "invalid" state, using a P-tag pointer.

As is apparent from the above description, the arrangements described provide a virtual address tag memory, together with the physical address tag memory, to the internal cache memory, thereby minimizing the time required in the address translation and thus improving the performance of the microprocessor.

Although preferred embodiments illustrative of the invention have been disclosed by way of example, those skilled in the art will appreciate that various

modifications, additions and substitutions are possible, without departing from the scope of the protection sought by the accompanying claims.

CLAIMS

1. A cache memory accessed by a virtual address from a microprocessor, the cache memory including a data memory for storing data in cache lines thereof, a virtual tag memory for storing a virtual address tag, wherein the virtual tag memory is accessed by a request from a microprocessor in parallel with the data memory, a physical tag memory for storing a physical address tag, wherein the physical tag memory is coupled to an address translation means and a bus monitoring logic circuit, and wherein the address translation means translates the virtual address into a physical address corresponding to the physical address tag, and a pointer associating the virtual address tag with the physical address tag.

2. A cache memory as claimed in claim 1, wherein the microprocessor includes a plurality of multiplexing means for copying state information in the physical tag memory into the virtual tag memory and updating the virtual tag memory and the physical tag memory.

3. A cache memory as claimed in claim 1, wherein the bus monitoring logic circuit makes the data invalid, and makes the corresponding items invalid in the virtual tag memory and the physical tag memory.

4. A method for accessing a c ache memory using a

virtual address in response to a request from a microprocessor, including the steps of storing data in the cache lines of a data memory, storing a virtual address tag in a virtual tag memory wherein the virtual tag memory is accessed by a request from a microprocessor in parallel with the data memory, storing a physical address tag in a physical tag memory wherein the physical tag memory is coupled to an address translation means and a bus monitoring logic circuit and wherein the address translation means translates the virtual address into a physical address corresponding to the physical address tag, and associating the virtual address tag with the physical address tag by means of a pointer.

5. A method for accessing a cache memory using a virtual address in response to a request from a microprocessor having a bus monitoring logic circuit, including the steps of determining whether the virtual tag corresponding to the virtual address is stored in a first memory, so that either a hit signal or a miss signal is generated, determining whether the state of a cache line in a first memory storing data is valid in the case where a hit signal is generated, performing a read or a write operation in the case where the state of the cache line is valid, translating the virtual address into a physical address in the case where the state of the cache line is invalid, determining whether a physical tag corresponding to the physical address is stored in a

second memory, copying state bits of the physical tag into the first memory if there exists a corresponding physical tag in the second memory and performing a read or a write operation, selecting entries to be replaced in the first memory in the case where there is not a corresponding physical address in the second memory or a miss signal is generated, determining whether there are entries in the second memory, which correspond to the entries to be replaced in the first memory, and, if there are not, determining the entries to be replaced in the second memory, re-selecting other entries to be replaced in the second memory and selecting the corresponding entries to be replaced in the first memory according to the re-selected entries, reading a new cache line from the external memory and performing a read or a write operation.

6. A method as claimed in claim 5, wherein the second memory is accessed by the bus monitoring logic circuit, using a physical address on a bus master.

7. A method as claimed in claim 6, wherein the bus monitoring logic circuit changes the state information of the physical tag and wherein the state information of the virtual tag is changed by the changed state information of the physical tag.

8. A method as claimed in claim 5, wherein the

step of reading the new cache line from the external memory includes the step of writing a modified cache line in an external memory.

9. A method as claimed in claim 8, wherein the external memory is an external cache memory.

10. A method as claimed in claim 8, wherein the external memory is a main memory.

11. A method as claimed in claim 5, wherein the first memory has pointer information which associates the virtual tag with the physical tag.

12. A method as claimed in claim 11, wherein the second memory has pointer information which associates the physical tag with the virtual tag.

13. A cache memory as claimed in claim 1 substantially as described herein with reference to any one of Figs. 4 to 6 of the accompanying drawings.

14. A method as claimed in either claim 4 or claim 5 substantially as described herein with reference to any one of Figs. 4 to 6 of the accompanying drawings.